

## DOCUMENT RESUME

ED J63 519

AA 001 001

AUTHOR Krahmer, Edward; Horne, Kent  
TITLE An Alternative to QUERY: Batch-Searching of the ERIC Information Collections.  
INSTITUTION Resource Information Center, Grand Forks, N. Dak.  
SPONS AGENCY Office of Education (DHEW), Washington, D.C.  
BUREAU NO BR-1-0673  
PUB DATE 1 Aug 72  
GRANT OEG-071-4134  
NOTE 55p.

EDRS PRICE MF-\$0.65 HC-\$3.29  
DESCRIPTORS \*Computer Programs; \*Data Processing; \*Information Retrieval; \*Information Systems; \*Manuals; Search Strategies  
IDENTIFIERS QUERY

## ABSTRACT

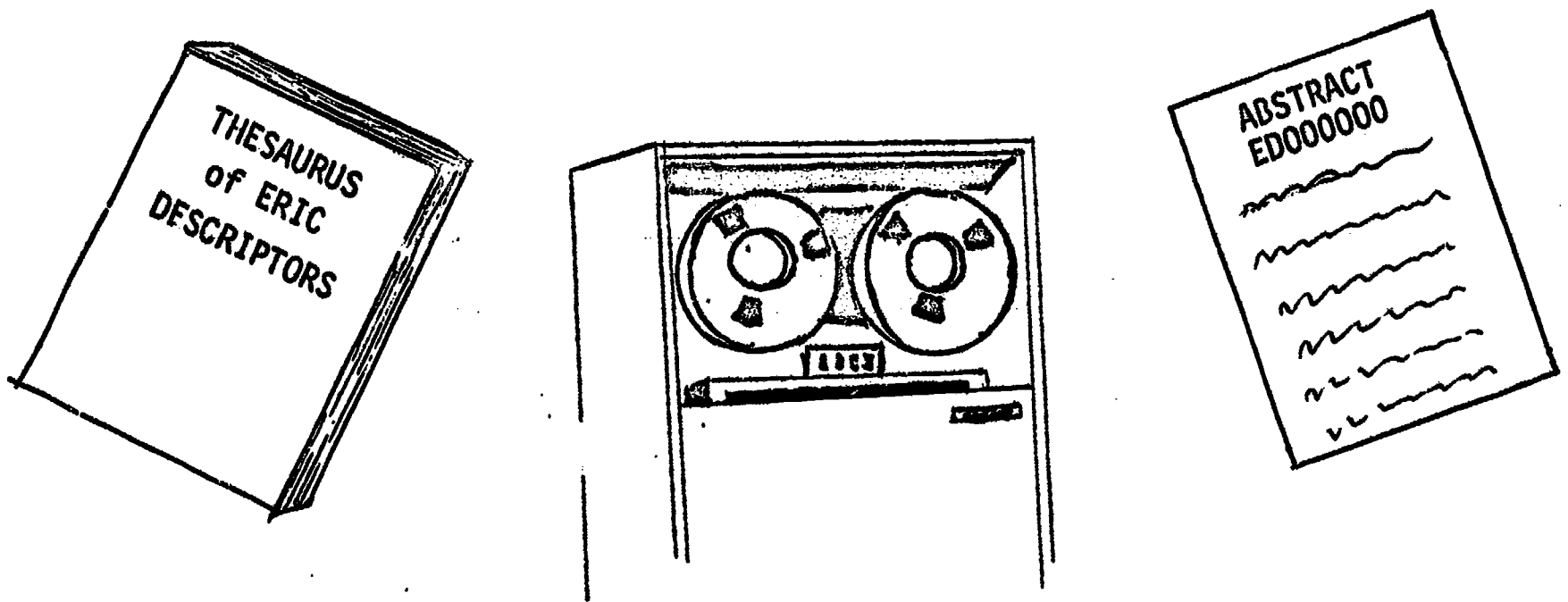
A manual describing the RIC computer search program for retrieval of information from ERIC, CIJE, and other collections is presented. It is pointed out that two versions of this program have been developed. The first is for an IBM 360/370 computer. This version has been operational on a production basis for nearly a year. Four installations of this version have been made by RIC for the State Educational Agencies in Iowa, Kansas, Massachusetts, and Texas. The second version is for mini-computers using the BASIC programming language. This version, still in the developmental stages, is operational only on a Digital Electronics Corporation PDP-12 time-sharing mini-computer. Numerous programming modifications to improve the performance of this version, as well as documentation of this program, are underway. A third version, a combination of the other two versions appears to be economically most attractive. This "hybrid" approach uses a 360/370 computer for performing the logic and the mini-computer for printing. The manual places emphasis on how to utilize the RIC program to perform searches; consideration of some technical aspects is given in the Appendices. The approach taken by RIC, utilization of the ERIC Descriptor Postings, provides one alternative to the computer search program, QUERY. (Author/CK)

FILMED FROM BEST AVAILABLE COPY

ED063519

## AN ALTERNATIVE TO QUERY

### BATCH-SEARCHING OF THE ERIC INFORMATION COLLECTIONS



AUTHORS:                   EDWARD KRAHMER  
                              KENT HORNE  
SYSTEM ANALYSTS:       LEE BRUENI  
                              JERRY BERG

Grant Number: OEG-071-4134  
Project Number 1-0673

RESOURCE INFORMATION CENTER  
GRAND FORKS, NORTH DAKOTA

AUGUST 1, 1972

AA001001

**AN ALTERNATIVE TO QUERY  
BATCH-SEARCHING OF THE ERIC INFORMATION COLLECTIONS**

**Prepared by:**

**AUTHORS:** Edward Krahmer  
Kent Horne

**SYSTEM ANALYSTS:** Lee Brueni  
Jerry Berg

**Published by:**

**RESOURCE INFORMATION CENTER  
Box 8009 University Station  
Grand Forks, North Dakota 58201**

**1971-72 Staff:**

**Co-Directors:** Edward Krahmer and Ronald Torgeson  
**Information Specialists:** Patricia Berntsen, Donald Eshelby,  
Kent Horne, and Sandry Kelly  
**Systems Analysts:** Jerry Berg and Lee Brueni  
**Secretaries:** Karen Heskin, Irene Wagner, and  
Jane White

**Under Contract to:**

**Title III of the Elementary and Secondary Education Act  
North Dakota Department of Public Instruction**

**Division of Extension Services  
National Center for Educational Communications  
United States Office of Education**

**The research reported herein was performed pursuant to support received from the U. S. Office of Education and the North Dakota Department of Public Instruction. Points of view or opinions stated do not necessarily represent the official position or policy of the sponsoring agencies.**

**August 1, 1972**

## T A B L E O F C O N T E N T S

FOREWARD . . . . .	V
CHAPTER I. . . . . ERIC and the Need for Computer Retrieval Capability	1
CHAPTER II . . . . . RIC and Its Computer Search Procedure	4
CHAPTER III. . . . . Coding Searches for Computer Input	11
CHAPTER IV . . . . . Hardware Configurations with Cost Analyses	21
CHAPTER V . . . . . Additional Consideration	30
APPENDIX A . . . . . Documentations for Options In RIC IBM 360/370 Computer Search Program	33
APPENDIX B . . . . . Additional Hardware Configurations and Costs for the Mini-Computer Program	47

## FOREWARD

The contents of this manual describe the RIC computer search program for retrieval of information from ERIC, CIJE, and other collections. Actually, two versions of this program have been developed. The first is for an IBM 360/370 computer. This version has been operational on a production basis for nearly a year. Four installations of this version have been made by RIC for the State Educational Agencies in Iowa, Kansas, Massachusetts, and Texas. The second version is for mini-computers using the BASIC programming language. This version, still in the developmental stages, is operational only on a Digital Electronics Corporation PDP-12 time-sharing mini-computer. Numerous programming modifications to improve the performance of this version, as well as documentation of the program, are underway at this time. However, a third version, a combination of the other two versions, appears to be economically most attractive. This "hybrid" approach uses a 360/370 computer for performing the logic and the mini-computer for printing.

Emphasis is placed in this manual on how to utilize the RIC program to perform searches; however, consideration of some technical aspects is given in the Appendices. While emphasizing how to code searches, the treatment in this short manual cannot obviously cover all possible circumstances. A user of the RIC computer search program should contact the authors for assistance with regard to any problem encountered which is not covered in this manual.

The approach taken by RIC, utilization of the ERIC Descriptor Postings, provides one alternative to the most widely-known computer search program, QUERY. After reading this manual, educators interested in obtaining further information about and/or installing the RIC computer search program should contact either of the authors.

The authors wish to thank the past North Dakota Title III State Coordinator, Vernon Eberly, and present Coordinator, Glenn Dolan, for the Title III support of RIC without which the development of the RIC search program would never have been attempted. The University of North Dakota Computer Center, Conrad Dietz, Director, deserves considerable credit for facilitating the development and operation of the IBM 360 version. The South Junior High School, Grand Forks School District, Computer Center, Walter Knipe, Director, established under a Title III ESEA grant from the North Dakota Department of Public Instruction, deserves credit for sponsoring the mini-computer development effort. Finally, the authors wish to thank the initiators of the four installations of the RIC computer search program, Alice Schafer of Mitre Corporation, Boston; Dorthy Mueller, Texas Department of Education, Austin; Richard Herlig, Kansas State Department of Education, Topeka; and Mary Jo Bruett, Iowa Department of Public Instruction, Des Moines, for their many helpful suggestions for improving the program.

## CHAPTER I

### ERIC AND THE NEED FOR COMPUTER RETRIEVAL CAPABILITY

Anyone familiar with American education realizes that it is undergoing a knowledge explosion similar to that occurring in the sciences. The contents of the explosion include curriculum materials, instructional media, research reports, program descriptions, training materials, and numerous other types of information.

Until a few short years ago (the mid sixties) this explosion in education threatened to get out-of-hand. However, ERIC made its appearance in time to forestall the threatened inundation.

Briefly, ERIC is a system of twenty clearinghouses, each responsible for searching out, reviewing, assigning terms descriptive of the content and abstracting pertinent fugitive literature in an important domain of education. Documents and their abstracts selected for inclusion in ERIC are forwarded to Central ERIC, a branch of the National Center for Educational Communications of the United States Office of Education. Central ERIC in turn submits the materials from all clearinghouses to a contractor who microfilms documents which are not copyrighted. Each document appears on one or more microfiche; a 4 by 6 inch sheet of microfilm which can hold up to 72 pages. The contractor also prepares a monthly publication, Research in Education, containing the abstract and bibliographic information for each document. This publication serves as an index to the ERIC collection using the descriptor terms assigned to the documents by the clearinghouses as the index terms. The clearinghouses also review the educational articles of over 550 journals. The articles are assigned descriptors so that they



can be indexed similar to the documents in RIE and the bibliographic information appears in a monthly publication titled Current Index to Journals in Education; however, microfiche copies of the articles are not available.

While any organization can subscribe to the indices and obtain copies of the ERIC microfiche, the ERIC system is not without its special problems. Most obviously, thousands of educators must be informed of this system and its potential uses. Educators must then be provided ready access to ERIC materials, microfiche readers, and microfiche printers in order to maximize the utilization of the source documents on microfiche. Most importantly, educators must be stimulated and motivated to develop professionalism similar to that of the scientist in order that they might continually seek the best education program possible. This includes the effort to remain current within a specialty, something too few educators can yet boast.

It is not the intent of this document to cover the basic problems of the ERIC system. The more narrow concern is focused on the rapidly expanding holdings of ERIC and CIJE. As of July 1, 1972, ERIC had indexed 54,390 documents and CIJE had indexed 45,271 articles and the respective yearly rates of expansion for these two collections were 12,329 and 17,671. Any individual desiring to conduct a search of the ERIC and CIJE holdings for information pertinent to a particular topic has a major task even with the monthly issues of RIE and CIJE, the yearly summary indices, the thesaurus of descriptor terms and assuming the person has access to these indices. As a means of lessening the burden on the individual, the computer is now being called upon to perform the clerical chores of searching the ERIC and CIJE collections. It is the function of this document to describe one approach to computerized searching of the ERIC information collections.



Before considering further the RIC computer search procedures, the reader should note several definitions. Document refers to any single entry into the ERIC information collections. It is even used to refer to an article in CIJE. Abstract specifically refers to the short description of the contents of each document. For articles in CIJE an abstract frequently does not appear. Resume refers to the abstract with appended bibliographic data, i.e., the entire entry for a document appearing in RIE or CIJE. Document, abstract and resume are used synonymously throughout this publication. Finally, abstract or resume numbers are also termed as accession numbers. An accession number is a six-digit number prefaced by an ED for documents in RIE and an EJ for articles in CIJE.

## CHAPTER 2

### RIC AND ITS COMPUTER SEARCH PROCEDURE

RIC (the Resource Information Center) was established September 1, 1969, through a contract with the North Dakota Department of Public Instruction using Title III ESEA administrative funds. During the first year of this contract, developmental work was undertaken by the Upper Red River Valley Educational Service Center. The two renewals of this contract have seen RIC operate from quarters on the University of North Dakota campus where ready access is assured to a complete ERIC microfiche collection in the ERIC Center of the University Library. RIC also is the recipient, commencing July 1, 1971, of a National Center for Educational Communications, U. S. Office of Education, grant for developing a "local educational information center." The Grand Forks School District serves as the contractor for this grant.

The purpose of RIC is to provide, as an arm of the North Dakota Department of Public Instruction, a comprehensive "one-stop" source of educational information for the educators of the State. RIC undertakes a number of activities intended to foster awareness and utilization of its services. These activities include inservice workshops, slide/tape presentations, brochures, a monthly Newsletter, and follow-up contacts with users. The ERIC and CIJE collections serve as the primary source of information, however, RIC also uses the University Library's holdings, ALERT (from the Far West Educational Laboratory) and a number of locally-developed collections. Special products of RIC include brief reports summarizing the literature on critical educational topics and announcement of PREP.

3

RIC places emphasis on media specialists (librarians) as its primary information contacts with the educators of the State. Training efforts focus on this group in that RIC's limited staff could not personally reach all educators. Regional centers, where microfiche reader/printers, copies of Research in Education, and RIC publications are available, have been established throughout the State within reasonably easy access to all educators until such time as their school districts provide microfiche readers for their use.

Efforts of RIC to secure computer search capability date back to 1969. At that time it was noted in one of the ERIC clearinghouse newsletters that a copy of the ERIC computer tape could be secured on loan from Central ERIC. RIC in turn made a copy of this tape and initiated an effort to develop a computer search procedure on an IBM 360/30 computer.

While progress was being made in reading the ERIC tape, RIC terminated development of this program later in 1969 when announcement of the QUERY search program was received from Central ERIC. RIC subsequently became the twenty-second installation of the QUERY program. Unfortunately it was quickly found that QUERY was uneconomical considering several factors including the limited capacity of the 360/30 with its very slow (18 1/2 inches per second) tape drives. (However, this was a blessing in disguise in that it halted what would have proved to be an equally uneconomical and costly development of software.)

Casting around for an alternate approach to securing an economical computer search capability, the RIC director was introduced by James Eller of Central ERIC to the publication, ERIC Descriptors - Term Usage Postings and Term Usage Statistics, produced by LEASCO, the computer contractor for ERIC. RIC was one of the first to order this publication. More importantly,

the nature of this publication, an alphabetic listing of ERIC descriptors with a numeric listing of all documents indexed by each descriptor, caused the RIC director to contact LEASCO with a request for a copy of the computer tape from which this publication was printed. (The first page of the descriptor postings appears on the next page.) RIC was the first non-ERIC agency to receive what is now called the Descriptor Posting Tape or USEMAST.

Reviewing the approach to searching the ERIC collection taken by QUERY and many of its modifications, the ERIC resume or document tapes are directly entered by the search program. Descriptor terms for each document are directly checked to see if they meet the criteria set by the logic for each search included in the batch of searches under consideration. Hits, or documents which meet the criteria, are stored on tape until a later time when they are sorted by search number and printed. While QUERY provides many nice features such as comparing on a descriptor prefix or suffix, this is at the expense of hundreds or even thousands of comparisons when it is considered that there are an average of 10.5 descriptors assigned per document and a batch of searches may contain a hundred or more descriptors. QUERY also frequently has a fairly complicated procedure for coding searches into computer usable form.

By limiting the search capability to the essential element, the ability to perform AND/OR/NOT logic, the use of the descriptor-posting tape can reduce the time required to perform a comparable batch of searches against the entire ERIC file to a fraction of that required by QUERY. Instead of entering each resume to check each descriptor in the batch of searches, the accession numbers of the resumes which are indexed by each descriptor are read from the descriptor-posting tape, or from random access disk for even greater time savings. The logic is then performed on the accession numbers

FIGURE 1

## A PAGE from the RIE Descriptor Postings or Inverted File

## DESCRIPTOR

## Abbreviations

## Ability

## Ability Grouping

## Ability Identification

## Able Students

## Abortions

## Abstracting

## Abstraction Levels

## Abstraction Tests

## Abstract Reasoning

ED021419 ED025274 ED034171 ED035717 ED049984 ED057630

ED001617 ED001731 ED002237 ED002573 ED002810 ED002860 ED002942 ED003190 ED003284  
 ED003294 ED003630 ED003683 ED010035 ED010186 ED010267 ED010549 ED010635 ED011140  
 ED011344 ED011345 ED011429 ED013762 ED013853 ED015678 ED015795 ED016060 ED016904  
 ED017190 ED017851 ED017940 ED018037 ED018115 ED018117 ED018326 ED018574 ED018713  
 ED018899 ED018998 ED019096 ED019719 ED019762 ED019882 ED020126 ED020284 ED020592  
 ED020677 ED021373 ED022023 ED022844 ED023469 ED023968 ED024160 ED025011 ED026117  
 ED027123 ED027583 ED027883 ED028931 ED028939 ED029340 ED031682 ED031742 ED032009  
 ED032124 ED032372 ED032944 ED033295 ED033484 ED034128 ED034217 ED034218 ED034219  
 ED034259 ED034274 ED035013 ED035051 ED035937 ED036774 ED037784 ED037805 ED039423  
 ED039438 ED039538 ED039546 ED039616 ED040462 ED041344 ED042122 ED042181 ED042771  
 ED042632 ED042807 ED042984 ED043087 ED043189 ED043769 ED043804 ED043893 ED043922  
 ED044425 ED044530 ED044548 ED044586 ED045258 ED045281 ED045282 ED045921 ED048330  
 ED048946 ED049046 ED049289 ED049861 ED049868 ED050374 ED050805 ED051286 ED051309  
 ED052352 ED052863 ED055289 ED059766 EP010121 EP010446 EP012017 EP000123

ED001063 ED001111 ED001115 ED001177 ED001198 ED001202 ED001207 ED001209 ED001214  
 ED001215 ED001219 ED001222 ED001227 ED001315 ED001372 ED001375 ED001381 ED001404  
 ED001452 ED001692 ED001742 ED001883 ED002036 ED002093 ED002101 ED002108 ED002114  
 ED002254 ED002320 ED002770 ED002939 ED003061 ED003249 ED003358 ED003476 ED003842  
 ED010003 ED010051 ED010210 ED010619 ED010701 ED011302 ED011344 ED011673 ED011816  
 ED011909 ED011738 ED013710 ED013842 ED014088 ED014812 ED015194 ED015678 ED015974  
 ED016603 ED017002 ED017048 ED017329 ED017425 ED017520 ED017885 ED017937 ED018003  
 ED018688 ED019120 ED019186 ED019291 ED019348 ED019512 ED019796 ED020219 ED020675  
 ED020878 ED020882 ED020983 ED021505 ED021506 ED021709 ED021864 ED021865 ED021915  
 ED022253 ED022659 ED023157 ED023295 ED025003 ED025316 ED025811 ED025842 ED025944  
 ED026135 ED026281 ED027009 ED027166 ED027232 ED028549 ED029365 ED029370 ED030188  
 ED031504 ED031536 ED032444 ED032637 ED032638 ED032951 ED033047 ED035154 ED036829  
 ED039613 ED040513 ED041092 ED041623 ED042173 ED042271 ED042848 ED043617 ED043692  
 ED044435 ED044525 ED045318 ED045780 ED046921 ED047077 ED047415 ED047861 ED048381  
 ED048382 ED048383 ED048384 ED048969 ED050094 ED050143 ED050187 ED050196 ED051330  
 ED051418 ED051419 ED051638 ED052228 ED052260 ED052282 ED052567 ED052633 ED053194  
 ED054537 ED055838 ED056056 ED056150 ED056330 EP011181 ES000566 ES001202 ES001637  
 ES001930 ES002150 ES002346 ES002573

ED001199 ED001211 ED001212 ED001215 ED001223 ED001234 ED001249 ED001276 ED001280  
 ED001295 ED001325 ED001450 ED001515 ED001538 ED001621 ED001643 ED001648 ED001702  
 ED001750 ED001752 ED001757 ED001758 ED001760 ED001847 ED001882 ED001885 ED001919  
 ED002093 ED002176 ED002177 ED002820 ED002822 ED002839 ED002861 ED002864 ED003043  
 ED003293 ED003595 ED003858 ED010088 ED010206 ED010238 ED010420 ED010502 ED010535  
 ED012166 ED012612 ED012613 ED013103 ED013411 ED013516 ED013518 ED014653 ED015033  
 ED015506 ED015783 ED016137 ED017004 ED017475 ED019119 ED020038 ED020086 ED020801  
 ED023232 ED023295 ED023469 ED023722 ED023768 ED024716 ED025207 ED025811 ED026116  
 ED026562 ED026872 ED027796 ED027801 ED028931 ED028936 ED032124 ED033138 ED033223  
 ED034073 ED034265 ED034353 ED034807 ED035730 ED037784 ED038681 ED038737 ED039577  
 ED040351 ED041063 ED041261 ED041417 ED041896 ED042927 ED043395 ED043683 ED043684  
 ED044427 ED044433 ED044745 ED045687 ED046976 ED047009 ED047231 ED048381 ED048703  
 ED049046 ED049326 ED049343 ED050145 ED050196 ED051419 ED051428 ED052228 ED052376  
 ED053156 ED053178 ED053333 ED053343 ED054340 ED055099 ED055259 ED056057 ED056219  
 ED057162 ED057222 ED057343 ED060178 EP011501 EP011891 EP012026 ES002132 ES002176  
 ES002269 ES100176 FR001037

ED001149 ED001164 ED001201 ED001205 ED001208 ED001214 ED001233 ED001165 ED001388  
 ED001515 ED001764 ED001779 ED001884 ED002102 ED002165 ED002982 ED003224 ED003286  
 ED003406 ED003438 ED010316 ED010317 ED010318 ED010533 ED010766 ED010998 ED011002  
 ED011238 ED011350 ED011857 ED011979 ED012246 ED012580 ED013516 ED013752 ED014308  
 ED014422 ED014475 ED014486 ED015123 ED015697 ED015878 ED016664 ED017019 ED017473  
 ED017866 ED017937 ED018504 ED018505 ED019076 ED019251 ED019315 ED019693 ED019918  
 ED020046 ED020288 ED021928 ED023609 ED024590 ED024605 ED026248 ED026281 ED030672  
 ED031544 ED032444 ED033112 ED036245 ED036515 ED037522 ED038664 ED042806 ED043285  
 ED043286 ED044076 ED045038 ED046321 ED048819 ED049684 ED051741 ED055543 ED055565  
 EP010004 EP011145 ES000653 ES000811 ES000951 ES002132 ES002232 ES002269 ES002270  
 ES002410 ES002416 MP000616

ED054042 ED055869 ED058128

ED003474 ED003776 ED003805 ED003806 ED003813 ED003844 ED010298 ED010681 ED010840  
 ED014229 ED014256 ED015003 ED016913 ED017282 ED017293 ED017303 ED017311 ED017747  
 ED020489 ED023409 ED025283 ED026083 ED027919 ED029651 ED029670 ED029671 ED033379  
 ED033722 ED039887 ED040724 ED042716 ED043798 ED045170 ED046465 ED047740 ED048899  
 ED048905 ED048906 ED049799 ED050739 ED050800 ED052803 ED056016 ED058883 ED059204  
 ED059412 ED059753 EP010922 EP011559 FR001187 FR001189

ED016651 ED022305 ED023220 ED024169 ED024466 ED024471 ED025088 ED026752 ED026800  
 ED027317 ED027513 ED028004 ED028174 ED028200 ED028378 ED029891 ED030254 ED032585  
 ED033930 ED034271 ED034732 ED035441 ED035620 ED036576 ED037234 ED039190 ED040176  
 ED040459 ED040767 ED040833 ED042261 ED042764 ED043084 ED043395 ED045626 ED046196  
 ED047897 ED048683 ED049002 ED049617 ED050025 ED050556 ED050569 ED051036 ED052158  
 ED055447

ED012132 ED016528 ED020018 ED021608 ED021645 ED026752 ED029715 ED042806

ED012132 ED013665 ED014320 ED017519 ED017693 ED017694 ED017695 ED017696 ED017697  
 ED017698 ED017699 ED019285 ED019346 ED019623 ED020010 ED020078 ED022305 ED022384  
 ED023214 ED023231 ED023542 ED023543 ED023725 ED024182 ED024325 ED024466 ED024558  
 ED025093 ED025205 ED026131 ED026950 ED028831 ED030538 ED032585 ED033854 ED034693  
 ED034756 ED034819 ED035496 ED035620 ED037462 ED040342 ED041619 ED042806 ED043659



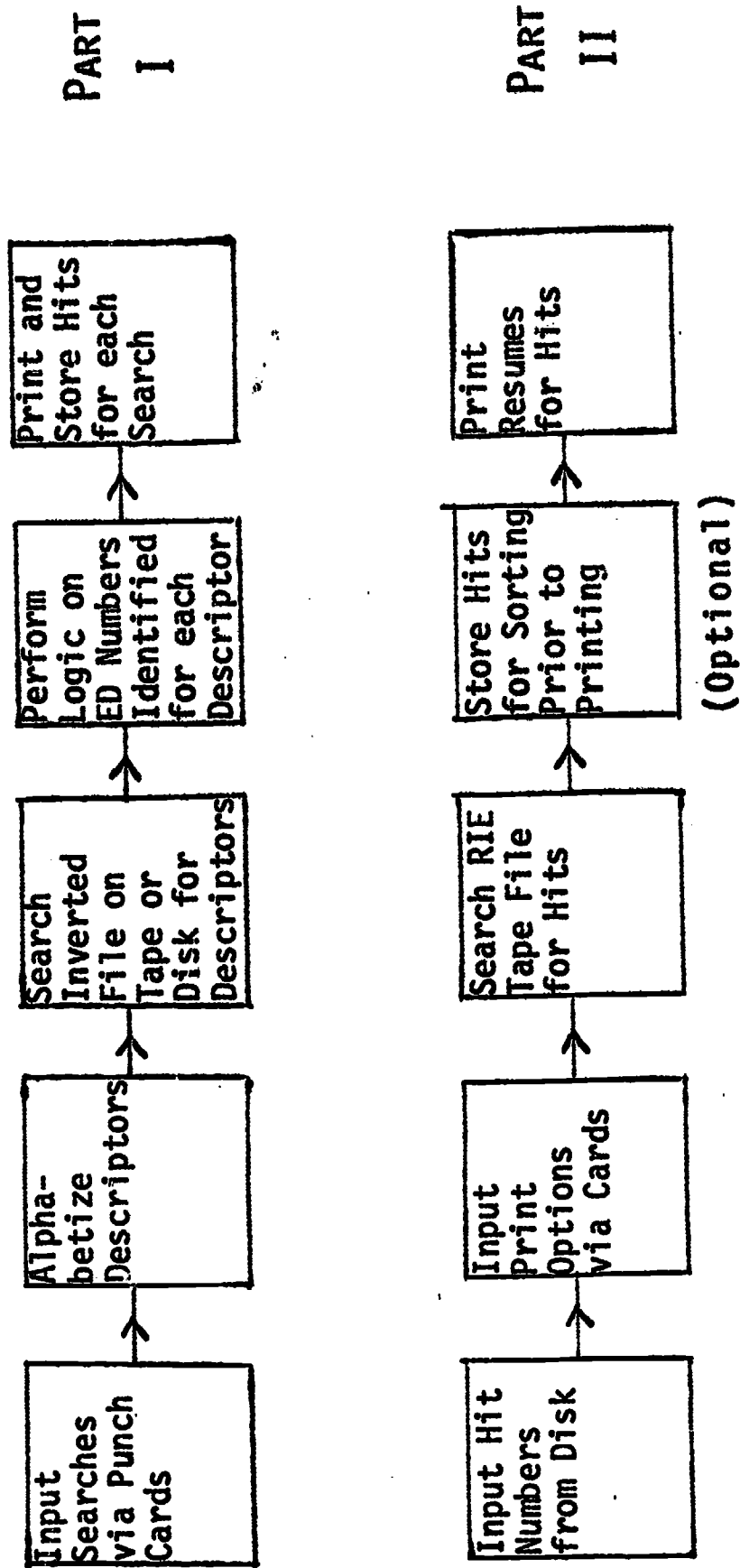
assigned each descriptor rather than on the descriptors themselves, a considerably faster process. The "hits" for each search are next sorted into numeric order. Finally the resume tapes are entered, but only to compare each resume number with the next number in the ordered list of hits. When a match is found, the result is either printed or stored on a work tape until such time as the hits can be sorted back into order by search number.

Figure 2 on the following page reveals that the RIC search program actually functions as two distinct parts. The first part accepts a batch of searches and alphabetizes the descriptors. Next, by searching the descriptor-posting tape for the alphabetized descriptors, a list of accession numbers, which contains all the possible hits for each search, is obtained. Then the logic assigned to the descriptors is performed. Finally, the resultant hits are both printed and stored on disk by search numbers.

Before entering the second part of the RIC search program, it is possible to review the hits to see if they fit the question asked. This is a distinct advantage over QUERY where the results are seen only after the resumes are printed. It is possible to enter a number of options providing for the printing of only those resumes desired. The second part of the RIC search program applies these options to the lists of hits stored on disk. The resume tapes are then entered to find the desired resumes. These resumes are either directly printed or stored on disk or tape for sorting in order by search number prior to printing.

Besides the printing options, a number of other options as to page size and length, number of resume tapes and accession numbers on each tape are built into the program. In Appendix A will be found copies of

FIGURE 2  
DIAGRAM OF RESOURCE INFORMATION CENTER COMPUTER SEARCH PROGRAM





the documentation for these options as the documentation appears in print-outs from the IBM 360 version of the program.

### CHAPTER 3

#### CODING SEARCHES FOR COMPUTER INPUT

Essential to utilizing any ERIC computer search program is an understanding of the techniques for coding searches. Figure 3 illustrates the elements of a search coded for the mini-computer and for the 360/370. version of the RIC computer search program. An understanding of only four elements is required to code searches: 1) search number, 2) end of search symbol, 3) descriptors, and 4) logic-parentheses. The first three elements are self-explanatory; only the latter element will be considered in this chapter.

Before detailing the coding procedures for the RIC computer search program, the reader should become thoroughly familiar with the Thesaurus of ERIC Descriptors. It is essential that anyone using the RIC computer search program, or any other search program, select terms from the Thesaurus. Spelling must be carefully checked since as simple a matter as leaving an "s" off of a descriptor will invalidate the particular search of which the descriptor is a part. The remaining searches in the batch, however, will be processed in the normal fashion.

#### Coding for the Mini-Computer Search Program

Assembling a search for mini-computer input is a very simple procedure. The coder should use paper marked off in 48-space lines for teletype input as shown in the example in Figure 3. A search may involve one or more lines of information. Descriptor terms may be broken in the middle of a word or may be entirely contained on a single line since spaces are ignored by the computer. However, if a word is broken, be sure that a hyphen (-) is not used as would be the case with printed matter.

**FIGURE 3**

**Examples of Search Coding Elements**

### Example for the Mini-Computer

[illegible]

### Example for the IBM 360/370 Computer

	1	10	20	30	40	50	60	70	80
Search Number									
End of Search Symbol									
Logic									
Descriptor									
Parenthesis									
	0001	ELEMENTARY EDUCATION .OR. ELEMENTARY SCHOOLS .AND. (EVALUATION .OR. MEASUREMENT)\$							

After each descriptor term one of the following five symbols must appear to tell the computer program how to process the descriptor:

- % apply OR logic
- & apply AND logic
- @ apply AND logic to two or more descriptors joined by OR logic
- # apply NOT logic
- \$ end of search. Must be used or the search will not terminate.

#### Coding for the 360/370 Computer Search Program

The 360/370 version of the RIC computer search program should be coded on standard 80-column data forms for punching IBM cards. The first four columns are reserved for the search number. A search may involve more than one line of information. Since spaces are ignored, descriptors may be broken at any point or contained entirely on a line. Again, do not use a hyphen (-) when breaking a word.

The following three logic symbols are used between descriptors:

.OR.

.AND.

.NOT.

The periods are essential.

The dollar sign (\$) is used to signify the end of each search.

#### Examples of Coding Logic

Several examples will be provided to demonstrate the application of these two search programs. The mini-computer search will appear on the left-hand and the 360/370 version on the right-hand side of the page. Descriptor terms will be assigned letters of the alphabet in these examples.

1. A % B \$

A .OR. B \$

Application of OR (%) logic results in the combining of posting lists

for these two descriptors. Duplicate numbers are retained only once.  
For example, the following postings would produce the results shown:

A ED000667 ED000984 ED001856

B ED000875 ED000984 ED001122

Result ED000667 ED000875 ED000984 ED001122 ED001856

Note that the dollar sign (\$) appears as it must as the termination of the search for either version of the RIC computer search program.

2. A & B \$ A .AND. B \$

Application of AND (&) logic results in only numbers common to the postings for both A and B being retained. In the example given above, only ED000984 is common to both descriptors. Thus, only one number would result from performing this search.

3. A % B & C \$ A .OR. B .AND. C \$

Complex searches can be written by simply combining the AND (&) and OR (%) logic of examples 1 and 2. It should be borne in mind that the program conducts the search from left to right. Thus, the computer reads the OR (%) symbol combining A and B as in example 1 before "seeing" the AND (&) logic symbol. It is the combination of A and B which the computer compares to C for numbers common to both as in example 2, rather than comparing B to C as might seem to be the case when looking at the example.

4. C & B % A \$ C .AND. B .OR. A \$

It will be noted that this is simply example 3 written in the reverse order. However, the result of applying the logic will not be the same. Note that when working from left to right the program first identifies numbers common to the postings for both C and B. Next it combines the common numbers with all the numbers in A. This is certainly not the same as for example 3.

5. C &amp; B &amp; A \$

C .AND. B .AND. A \$

On first glance this might appear to be the answer. However, working from left to right reveals that the numbers common to C and B are compared to A for numbers also common to it. The final results are numbers common to the postings for all three descriptors.

What is required is in effect a set of parentheses around the latter two descriptors such as C & ( B % A ) \$

6. C @ B % A \$

C .AND. (B .OR. A) \$

The @ symbol serves to indicate the presence of a set of parentheses in the mini-computer version; the 360/370 program accepts the actual parentheses up to five levels. Two or more descriptors connected by the OR (%) symbol always follow the @ symbol. The @ symbol signifies that the postings for each of these two or more descriptors will be compared using AND (&) logic to the postings of the descriptors preceeding the symbol. The parentheses for both versions of the RIC computer search program are terminated by another AND (&) or a dollar sign (\$) symbol. In the example, first B and next A are compared to C using AND (&) logic.

7. A % B @ C % D \$

A .OR. (B .AND. (C .OR. D)) \$

This example is comparable to the previous one with the exception that the postings for descriptors A and B are combined into one list first before reaching the AND (@) symbol.

8. A @ B % C &amp; D \$

A .AND. (B .OR. C) .AND. D \$

In this example, only those numbers in A which are either in B or C are retained since the @ symbol operates like a set of parentheses used in the 360/370 version. Remember that @ terminates when an & or \$ symbol is reached; in this example, the & symbol after descriptor C. What now happens is that the numbers which have been retained so far are compared

to the postings for descriptor D and only those numbers common to both lists are retained as hits.

9. A % B % C & D & E & F \$ ((A .OR. B .OR. C .AND. D) .AND. E)  
.AND. F \$

In this example the postings for descriptors A, B and C are combined into one list. Next, matches between this list and D are sought and written as a new list. This new list is compared in turn to E for matches and another new list is written. Finally, this list is compared to F for matches and the final hits are identified.

Now this example is somewhat artificial in that rarely does a search involve over two levels of AND logic. The obvious reason is that numbers common to more than two posting lists usually do not exist unless the descriptors are practically identical in which case only one of them would normally be used. The main point of this example is that no parentheses are required for AND logic in the mini-computer version while they are for the 360/370 version.

10. A % B % C & D % E & F \$ ((A .OR. B .OR. C) .AND. D) .OR. E .AND. F \$

This example is identical to the preceeding one except for the OR (%) logic symbol following descriptor D. However, this % creates the most serious coding problem for the mini-computer version yet encountered since there are two other ways in which this example could be written, each of which results in different hits.

10a. A % B % C @ D % E & F \$ ((A .OR. B .OR. C) .AND. (D .OR. E))  
.AND. F \$

10b. A % B % C & D \$ and ((A .OR. B .OR. C) .AND. D) .OR.  
E & F \$ (E .AND. F) \$



For all three examples the postings for the first three descriptors are combined into a single list. Now consider in turn what each of the three examples will do next.

For example 10 those numbers common to D and the combination of A, B and C will be combined with the entire posting list for E and compared to F for common numbers.

For example 10a those numbers common to the merger of D and E and the combination of A, B and C will be compared to F for common numbers.

For example 10b, those numbers common to D and the combination of A, B and C will be considered to be hits. Also the numbers common to E and F will be considered to be hits. In effect, two separate searches have resulted for the mini-computer version, but under the same search number. (It is possible that duplicate abstracts would be printed by this approach, however, a simple modification to the programs would remedy this situation if it proves to be a serious problem.)

Consider the following simple example to see what happens. (Hits are indicated by the boxes.)

Descriptor	Postings	Example 10	Example 10a	Example 10b
A	1,5			
B	2,4	1,2,4,5	1,2,4,5	1,2,4,5
C	3,4	1,2,3,4,5	1,2,3,4,5	1,2,3,4,5
D	0,2,3,6	2,3	(2,3)	<span style="border: 1px solid black;">2,3</span>
E	1,7,8	1,2,3,7,8	2,3,4	
F	1,2,5,8	<span style="border: 1px solid black;">1,2,8</span>	<span style="border: 1px solid black;">2</span>	<span style="border: 1px solid black;">1,8</span>

As is very apparent, the coder must be sure to understand exactly what it is he wishes the OR (%) to do. Even more important the coder

must understand the search request which he is attempting to code. Remember that the program processes from left to right. When coding a search, it is recommended that you think from left to right as the computer will do in order to determine whether the correct logic symbols have been assigned to the search.

11. A % B # C \$ (A .OR. B) .NOT. C \$

This brings us to the third type of logic, NOT (#). The NOT logic means everything will be processed except what is in the descriptor or set of descriptors following the NOT. Consider the postings listed below.

A	ED 000567	ED 001123	ED 001535
B	ED 000967	ED 001197	ED 001268
C	ED 000868	ED 001123	ED 001268

Note that combining A and B results in six numbers, two of which are contained under C. When the NOT logic is applied, these two are dropped and the result is:

ED 000567	ED 000967	ED 001197	ED 001535
-----------	-----------	-----------	-----------

The NOT logic can be used as many times as is desired in a given search.

#### Mini-Computer Summary

It will be noted that the mini-computer version of the RIC search program is not quite as flexible as the IBM 360/370 version. Multiple levels of parentheses, coding of searches entirely in English language, and sorting of abstracts in order by search number are all not possible at this stage in the development of the mini-computer version. However, the mini-computer program does have the capability to enter searches at any time; a batch of searches then consists of all searches entered since the last run of the logic program.

Particular attention should be given to the limited parentheses capability. Only one level of parentheses is possible using the @ symbol. This symbol is used when it is desired to combine two or more descriptors and compare the results with one or more other descriptors using AND logic.

It should also be noted that the mini-computer version still requires considerable programming. For example, additional levels of parentheses, more print options, rewriting portions of the program in assembler language, and several other revisions should result in an even more cost-effective program.

#### IBM 360/370 Version

The 360/370 version is a more complete and better documented (see Appendix A) program. The program has an especially good set of diagnostic error messages. The only area requiring additional emphasis would appear to be the matter of parentheses. As Figure 4 on the following page indicates, there are two specific purposes for utilizing parentheses. First, parentheses are used to group sets of descriptors when two or more AND or NOT logic terms are used. A set of parentheses is required around each pair of individual or sets of descriptors connected by an AND or NOT logic term. The second use is to indicate sets of descriptors which are connected by AND or NOT logic. It is important to check that the same number of right and left hand parentheses have been used.

## FIGURE 4

## Two Uses of Parentheses in the IBM 360/370 Version

1. TO INDICATE A GROUP OF TERMS WHICH ARE TO BE ANDED OR NOTED TO ANOTHER TERM OR GROUP OF TERMS.

CORRECT: A .AND. (B .OR. C)

INCORRECT: A .AND. B .AND. C

A .AND. B .OR. C

THE FIRST INCORRECT METHOD IS TOO RESTRICTIVE. IT REQUIRES ALL THREE TERMS TO BE PRESENT BEFORE A HIT RESULTS.

THE SECOND INCORRECT METHOD PERFORMS A .AND. B CORRECTLY; HOWEVER, C IS ORED TO THE RESULT. THEREFORE, ALL THE POSTINGS UNDER C ARE ADDED TO THE HITS FOR A .AND. B.

2. TO GROUP THE TERMS WHEN TWO OR MORE ANDs ARE PRESENT. THIS IS AN IDIOSYNCRACY OF THE PROGRAM; NOT SOMETHING WHICH IS OBVIOUS.

INCORRECT: A .AND. B .AND. C

CORRECT: A .AND. (B .AND. C)

CORRECT: (A .AND. B) .AND. C

NOTE THAT THESE PARENTHESES CAN BE PLACED ANYWHERE. IF THREE OR MORE ANDs APPEAR IN RARE INSTANCES, NOTE THAT SETS OF PARENTHESES ARE REQUIRED WITHIN PARENTHESES.

A .AND. (B .AND. (C .AND. D))

COMBINATIONS OF THESE TWO USES OF PARENTHESES OFTEN OCCUR.

A .AND. (B .AND. (C .OR. D))

A .AND. ((B .OR. C) .AND. (D .OR. E))

THE IMPORTANT CAUTION TO NOTE, ALWAYS CHECK THAT THE SAME NUMBER OF LEFT AND RIGHT PARENTHESES ARE USED.

## CHAPTER 4

### HARDWARE CONFIGURATIONS WITH COST ANALYSES

Hardware configurations will be considered under two headings: Configurations for the IBM 360/370 and Configurations for the Mini-Computer.

#### Configurations for the IBM 360/370

When confronted by the unrealistic operational costs of the original QUERY program, the Resource Information Center initiated an effort to secure an alternative to QUERY. The RIC director had previously considered the possibility of batching individual searches to be run against the inverted file computer tape, which is actually the magnetic tape for preparing the familiar descriptor postings. The initial set of RIC programs consisted of segments in Fortran to perform the logic and in Assembler to read the inverted file and abstract tapes. As time allowed, the Fortran segments were rewritten in Assembler resulting in a considerable cost savings and improved operating flexibility.

The basic operating specifications for these programs revolve around breaking down the logic for each search in the batch into its component parts and processing the results against all the ERIC and/or CIJE postings for each of the descriptor terms used in the search logic. At that point a listing of the "hits" resulting from each requested search is printed. This listing of hits for each search may be then

utilized in several ways including: 1. generating ERIC abstracts with equipment such as the Remington Rand REMCARD, and 2. editing prior to abstract generation, including limitation of abstract generation to either current or historic information.

Being extremely efficient and flexible, the RIC search programs can be installed on any of the IBM 360 or 370 computers from a 64K model 30 upwards. Since the IBM supported sort routine is incorporated as an operational part of the RIC search and retrieval strategy, and since this program is the largest user of core memory, it may be safely assumed that any IBM 360/370 series computer capable of supporting the IBM sort routine will also be capable of supporting the RIC search strategy.

#### The Dedicated IBM 360 or 370 System

A dedicated 360 or 370 system would tend to be least expensive when using the smallest computer, i.e. 360/30. The 360/30 CPU, for example, is apparently able to handle the searches at a lower cost, however, a much slower speed, than the 360/40, /50, /65, etc., which are much faster, yet more costly, given the same tape and disk drive speeds and capabilities. In almost all cases the greatest economy of operation will result from use of the highest speed tape drives and the largest IBM 2314 disk drives. At least one tape drive and two disk drives are recommended; however, a modification of the program is available which can operate on magnetic tape alone. In this case at least three tape drives are recommended.



### Time Sharing and the IBM 360 or 370 System

The time sharing configuration is the most difficult for which to determine a least cost analysis. It is assumed at this time, based upon data obtained from the IBM manuals, that the largest of the IBM 360 or 370 computers should be used for the least cost time sharing operation of the RIC computer search program. Ideally the same 2314 disk drives and the highest speed, highest density, tape drives, should be available in the same numbers as for the dedicated system.

### Cost Analysis for the IBM 360 or 370 System

Providing cost analysis data is a complicated proposition because of the many factors involved. For example, internal to the program such factors must be considered as: number of searches in a batch, number of accession numbers or postings to be processed, number of "hits" obtained, number of hits to be printed, etc. External to the program such factors must be considered as the model of the computer and the billing procedure and cost.

An example will be considered having 35 search requests resulting in 2950 "hits." This is a medium sized batch since the number of searches in a batch can range from one to over a hundred depending upon the computer model and the factors internal to the program. It took 10 minutes and 46 seconds for the logic part of the program to be performed on an IBM 360/40. At a billing cost of \$55 per clock hour this part of the program was run for \$0.28 per search request. At this point users with a REMCARD system would terminate the program. Most users, however, would utilize computer printing of the resumes. For printing the 2950 hits, 112 minutes were required



using the slowest IBM tape drives. Thus, for an additional \$2.93 per search the requesters obtained the printed abstracts sorted--in order by search number.

Note that if billing was based on CPU rather than clock time the cost per search would likely be less depending upon the rate charged for CPU time. Also note that a fast tape drive would reduce the cost per search proportionately.

#### Configurations for the Mini-Computer

The development of the RIC computer search program was also undertaken on a Digital Electronics Corporation (DEC) PDP-12 computer (Basically the PDP-12 is nothing more than a PDP-8 with analog capability). The only significant departure from the IBM 360/370 version is that presently no computer sorting of resumes by search number is possible. The particular computer used (South Junior High School, Grand Forks Public Schools, as part of a Title III ESEA project) has disk, tape, and printer hardware as well as time-sharing capability. However, this amount of hardware is not essential as has been proven and will be described in Appendix A under several of the hardware configurations to be considered. Also, this program should operate on any of the many mini-computers supporting the BASIC language with but minor modifications.

The computer search program operating on the time-sharing PDP-12 computer will be described in this chapter; the following four additional configurations will be described in Appendix B.

- a. Minimal stand-alone system.
- b. Ideal stand-alone system.

c. Addition of hardware to minimal existing system.

d. Addition of hardware to ideal existing system.

For each of these configurations the costs for equipment and maintenance were taken from the latest DEC price list. Some cost savings could be realized by going to peripherals supplied by independent companies; however, more problems with maintenance and software should be anticipated.

The cost analysis figures are based on the following hypothetical situation:

1. Amortization of the computer system over a five year period.

2. One hundred (100) search requests a month; 6,000 over the five-year period.

3. Searches run in batches of ten for each operation of the program.

4. An average of 100 resumes printed per search, each resume averaging 35 lines of 60 characters in length.

5. The terminal time required to process an average batch under stand-alone operation is 382 minutes when a disk is available and 465 minutes when only magnetic tape is available. Under time sharing with five other users the terminal time when a disk is available is 408 minutes while with only magnetic tape it is 502 minutes. All time estimates are based on the further assumption that a 350 line per minute printer and a 45 inch per second 800 bits per inch IBM compatible magnetic tape drive are available.

6. One shift results in 40 hours per week for 50 weeks per year, resulting in 10,000 hours of terminal time over the five-year period. Running 600 batches of 10 searches in that time in stand-alone

operation with disk available requires 38.2 per cent of the terminal time while with only magnetic tape requires 46.5 per cent of the terminal time. Under time sharing the respective percentages are 40.8 and 50.2.

The reader will note that cost per search figures are given for both shared and dedicated use of the respective systems presented in this chapter and in Appendix B. (It might be possible in many educational settings to secure access to a mini-computer during evening and nighttime hours when it is not normally being used; resulting in a cost less than even that for shared operation.) It must also be emphasized that these cost per search figures cover only computer expenses; the time required to code the search for the computer and the time required to process the resultant resumes following printing by the computer are not included.

The time-sharing configuration is the hardest for which to present cost analyses since, as with the 360/370 version, so many factors enter into the picture. First is the factor of how many users are served; the time sharing PDP-8 can handle up to 16 (32 with some minimal additional hardware). The assumption will be made that the system is handling 6 users, each utilizing the system 40 hours per week. The second factor is how many 4k memory partitions are available; the more available the more users can be utilizing the system simultaneously. The assumption will be made that 2 partitions are available. Third, the magnetic tape drive cannot presently be operated under time sharing; it is assumed that time sharing is shut-off during the second shift and that the dedicated use of the system does not add additional cost to the computer search application.

41

\$107,500	Initial hardware costs
32,500	Maintenance contract for 57 months
72,000	Two shift operator (\$600 per month/operator)
10,000	Overhead
<hr/>	
\$222,000	Total
+ 24,000	Lease of six teletype units
<hr/>	
\$246,000	Total
x .1667	One sixth cost of total system
<hr/>	
\$ 37,000	Total cost for total operation of terminal
x 40.6	Percentage of terminal time required to run searches (see point six above)
\$ 16,630	Total cost for computer time
÷ 6,000	Number of searches over the five year period
<hr/>	
\$ 2.80	Computer time cost per search
.30	Cost for paper on which the resumes are printed
<hr/>	
\$ 3.10	Total cost per search on a shared basis

Assuming a dedicated terminal were to be required the cost analysis would be as follows:

\$222,000	Total cost of configuration e.
x .1667	One-sixteenth cost of total system
<hr/>	
\$ 37,000	Total cost for computer time
+ 4,000	Lease of teletype unit
<hr/>	
41,000	Total
÷ 6,000	Number of searches over the five year period
<hr/>	

\$ 6.83 Computer time cost per search  
+ .30 Cost for paper on which the resumes are printed

---

\$ 7.13 Total cost per search on a dedicated basis

Obviously the cost per search figures will fluctuate with the number of users; the eventual addition of sixteen users paying for the hardware will reduce the cost per search in half. If the magnetic tape units can be operated under time sharing, additional cost reductions should be possible. Finally, the refinements of the mini-computer version now underway will result in further reduction in the cost for performing searches.

An interesting approach is to combine the two computer program versions assigning to each the task it does most cost-effectively. The 360/370 version would be assigned the task of performing the logic while the mini-computer would be assigned the task of printing. The result is a cost per search of less than \$2.00; significantly better than either version of the program can do alone.

In conclusion the reader should note that many factors must be taken into account when estimating computer costs. Several factors which have not been considered will now be covered briefly. First, what is the optimum sized batch for most economical operation? Number of descriptors, number of postings for the descriptors, type of logic used, number of hits to print, etc., are all variables to be considered. The cost per search actually drops in logarithmic fashion, i.e., the cost for a batch of one or two searches is very high but the cost levels off fast until after twenty or more searches very little savings result. Second is the number of searches run. The figure 6,000 has no special meaning; actually it would appear that more searches would result in

a lower cost per search. Third is the five-year amortization period; there is no reason why this period could not be longer or shorter resulting in varied cost per search estimates. Fourth, 100 resumes per search will be considered excessive by many. By appropriate choice of logic and descriptors a reduced number of hits is possible resulting in significant time and cost savings. Fifth, greater or lesser utilization of the computer system than the 40-hour week will result in appropriate modifications in the cost per search figures.

## CHAPTER 5

### ADDITIONAL CONSIDERATIONS

This concluding chapter will review briefly the following three considerations:

1. Printing resumes by search number.
2. Creation of inverted files as input to the logic program.
3. Inclusion of locally-generated information collections to be searched.

#### Printing Resumes by Search Number

A consideration the user of this computerized ERIC search procedure must make is the printing of resumes in order of accession numbers or in order of search numbers. The first option results in the necessity to hand sort the resumes--a time-consuming but usually less expensive proposition than computer sorting.

Users who feel the expense of computer sorting warranted have three obvious alternatives. First, anyone having access to the QUERY program might use the QSORT portion of QUERY. It should be possible to modify QSORT, however, the authors of this report have not tried this modification. The second alternative is to store the output from the abstract printing program onto a work tape or disk instead of going to the printer. A simple routine can then be used to read the work tape or disk once for every search, printing only those resumes numbered the same as the search being processed. The third alternative also involves writing the output on tape or disk. However, the output is then sorted on three dimensions, 1) search number, 2) accession number, and 3) line number of the material



to be printed. It is then possible to print the output in order by search and accession numbers. This modification has been tried and appears to be the most cost effective--only costing approximately 30 cents per search when processing a batch of 70 searches with 5,000 hits to be printed.

### Creation of Inverted Files

Creation of an inverted file, such as USEMAST supplied by LEASCO for the ERIC resumes, is not an unduly complicated concept or task. An inverted file is simply a one-dimensional search for every term in a selected field with the results being alphabetized or otherwise ordered. For example, for USEMAST the "given field" is the descriptor field. For each descriptor all the documents which are indexed by the descriptor are identified and the accession, or ED, numbers are listed after the descriptor as if a one-dimensional search had been performed. The descriptors are then alphabetized and the result is USEMAST.

Users interested in special applications of the ERIC data base, such as the ability to perform searches by institution, author, etc., can create inverted files for these fields. The results can be added to USEMAST so that the user could, for example, search for all the special education (descriptor) materials produced by a given institution of higher education.

### Locally Generated Information Collections

Another possibility available to the user of this ERIC search program is the inclusion of other information data bases other than ERIC and CIJE. The requirement for adding data bases to the system is solely that the information must be in computer readable format, i.e., on magnetic tape or disk. The user then turns to the inverted file generating program described in the proceeding paragraph and creates an inverted file on the one or more fields of interest for searching purposes.

The input data for the fields of interest being utilized for development of locally-generated inverted files must be consistent throughout that entire file. For example if an inverted file were to be generated for year of publication, then all input data must be consistent in having a four-digit number for year of publication (1950, 1951, etc., not 1950-51, 51, etc.).

Once the inverted file is created, the RIC logic program can be used to search for information. Identical coding procedures are followed as were shown in Chapter 4. The results can be printed using the resume printing program; however, some limited modifications might be required in order that the user can specify exactly what it is desired to print.

The Resource Information Center has developed a general purpose program for generating inverted files on most any information data base. Further information about this program and its utilization can be obtained by contacting either of the authors of this publication.

## APPENDIX A

### DOCUMENTATION FOR OPTIONS IN RIC IBM 360/370 COMPUTER SEARCH PROGRAM

The user of the RIC computer search program should find it to be quite well-documented. The programmer has included a brief explanation for every major section to describe exactly what the program is doing at that point.

Actually the "program" consists of the following five programs in two parts; the first performing the logic and the second printing the "hits."

PART	PROGRAM NAME
I Logic	DATETIME
	ERICCARD
	ERICUSE
	ERICLOG
II Printing	ERICABST

The documentation occurring at the beginning of each of the five programs consists of the following elements:

1. Identification of the program.
2. Function: a brief description of what the program does.
3. Program Options (optional) describing what user specified options are available.
4. Entry Points.
5. Macros Used.

6. Data Sets Used.

The documentation for each of the five programs will be found for reference purposes on separate pages headed by the program names.

Program Name: DATETIME

Identification

System	- ERIC Retrieval System
Program Name	- DATETIME
Sponsor	- Resource Information Center
Programmer	- Lee Brueni
Installation	- U of North Dakota Computer Center
Machine	- IBM 360/40G
OP System	- OS/PCP
Date Written	- May, 1972

Function

Obtain and return to the Calling Program the date and time in edited form (If. MM/DD/YY, HH.MM.SS).

Entry Points

DATETIME	- Entry Point to Program
----------	--------------------------

Macros Used

Time  
Save  
Return

Program Name: ERICCARD

### Identification

System	- ERIC Retrieval System
Program Name	- ERICCARD
Sponsor	- Resource Information Center
Programmer	- Lee Brueni
Installation	- U of North Dakota Computer Center
Machine	- IBM 360/40G
OP System	- OS/PCP
Date Written	- December, 1971

### Function

Read and parse each search request and construct descriptor records to be used in searching the usage master data set (inverted file). The table of descriptors is sorted into alphabetic order prior to being written to output set RICWORD. A brief summary of the number of searches processed and number of descriptors processed is also generated.

If an error is detected in a search request an error message is printed, and the search in error is deleted from the descriptor table.

### Program Options

Pagesize	- Change the number of lines per page to a new value from the default of 60.
Col. 1 - 8	- 'Pagesize'
Col. 9 - 10	- NN a number specifying the new page size

### Entry Points

ERICCARD	- Entry Point to Program
EOFIN	- End of File Branch Point for RICIN

### ^ :cros Used

Open  
Close  
Get  
Put  
Call  
Save  
Return



### Data Sets Used

RICIN

- This data set contains the control options to the program. Normally this is the card reader.

RICLIST

- This data set contains the printed list of the search requests. Errors detected in the search requests, and a summary report indicating the number of searches and descriptors processed, are provided.

RICWORD

- This data set is used to pass the sorted table of descriptors with logic, search number and word number to the following programs in the retrieval system. The format of the records is defined by -FMTWORDS-.

Program Name: ERICUSE

### Identification

System	- ERIC Retrieval System
Program Name	- ERICUSE
Sponsor	- Resource Information Center
Programmer	- Lee Brueni
Installation	- U of North Dakota Computer Center
Language	- Assembly F
OP System	- OS/PCP
Machine	- IBM 360/40G
Date Written	- December, 1971

### Function

Read the descriptor records created by the ERICCARD Program, then search for the descriptor in the usage master data set (also called the inverted file). When the descriptor is found, construct a record composed of information from the descriptor record created by ERICCARD and the accession number, then write the record into the accumulation data set.

### Program Options

Pagesize	- Change the number of lines per page to a new value from the default of 60.
Col. 1 - 8	- 'Pagesize'
Col. 9 - 10	- NN a number specifying the new page size
ACCTYPE	- Define the type of accession numbers that will be allowed to be processed. If this option is not chosen the default will be to allow processing of only ED type of accession numbers. Note that this option may be used more than once to chose more than one of the options.
Col. 1 - 8	- 'ACCTYPE-'
Col. 9 - 10	- XXX where XXX is replaced by one of the following choices.

ED - ED type of accession numbers  
will be accepted. (RIE  
collection)

EJ - EJ type of accession numbers  
will be accepted. (OIJE  
collection)

VT - VT type of accession numbers  
will be accepted. (Vocational  
Education)

ALL - No check will be made as to  
type of accession numbers to  
be processed.

### Entry Points

ERICUSE  
EOFUSAGE  
EOFWORDS  
EOFIN

- Entry Point to Program
- End of file branch point for RICUSAGE
- End of file branch point for RICWORDS
- End of file branch point for RICIN

### Macros Used

Open  
Close  
Get  
Put  
Read  
Check  
Note  
Point  
Call  
Save  
Return

### Data Sets Used

RICIN

- This data set contains the control  
options to the program. Normally  
this is the card reader.

RICLIST

- This data set contains the printed  
list of descriptors searched for,  
the search number and word number  
of the descriptor plus a count of  
the number of accession

numbers found. If an error occurred in locating a descriptor it is written into this data set. A summary of the number of descriptors and accession numbers processed is also generated.

#### **RICWORD**

- This data set contains the descriptors to be searched for plus the search number, word number, and logic. The format of the records is defined by the DSECT 'FMTWORDS'. This data set is created by the ERICCARD Program.

#### **RICUSE**

- This data set is commonly referred to as the 'inverted file' and contains the accession numbers listed under each descriptor. To enable processing a given descriptor more than once, data set positioning is used. To accomplish this the following I/O macros are used . . . Read, Check, Note, and Point.

#### **RICACCM**

- This data set is used for the accumulation of the hit records created from the merger of the accession numbers from 'RICUSE' with the search number, word number, and logic operators for a given descriptor from 'RICWORD'. The record layout is given by 'FMTACCM'.

Program Name: ERICLOG

### Identification

System	- ERIC Retrieval System
Program Name	- ERICLOG
Sponsor	- Resource Information Center
Programmer	- Lee Brueni
Installation	- U of North Dakota Computer Center
Machine	- IBM 360/40G
OP System	- OS-PCP
Date Written	- December, 1971

### Function

Read the logic records from the RICACCM data set created by the ERICUSE Program and insert all logic records of the same search number and accession number into the logic table (LOGICTAB). When the logic table is complete, then scan through the table performing the logic operations level by level, starting at the inner most, working outward. If a hit results then write it to the RICHOLD data set.

Upon completion of processing of the RICACCM data set, the data set RICHOLD is processed as input creating the data set RICHITS and printing the search hit report. The records written to RICHITS contain the search number, accession number, plus current and historic counts which are used by the abstract printing program ERICABST.

### Program Options

Pagesize	- Change the number of lines per page to a new value from the default of 60.
Col. 1 - 8	- 'Pagesize'
Col. 9 - 10	- NN a number specifying a new page size.

### Entry Points

ERICLOGC	- Entry point to program
EOFACCM	- End of file branch point for RICACCM
EOFHOLD	- End of file branch point for RICHOLD
EOFIN	- End of file branch point for RICIN

### Macros Used

Open  
Close  
Get  
Put  
Call  
Save  
Return

### Data Sets Used

RICIN

- This data set contains the control options to the program. Normally this is the card reader.

RICLIST

- This data set contains the printed list of all hits plus a summary hits list. The normal output device is a line printer.

RICACCM

- This data set contains records composed of a search number, accession number, word number and logic which is to be operated on. The format of the records is given by FMTACCM.

RICHOLD

- This data set is used to temporarily hold the hits records before writing them to the hit data set RICHITS. The format of the record is given by FMTHOLD.

RICHITS

- This data set contains the final hit records with the current and historic numbers assigned. The format of the record is given by access.



Program Name: ERICABST

### Identification

System	- ERIC Retrieval System
Program Name	- ERICABST
Sponsor	- Resource Information Center
Programmer	- Lee Brueni
Installation	- U of North Dakota Computer Center
Language	- Assembly F
OP System	- OS/PCP
Machine	- IBM 360/40G
Date Written	- August, 1971

### Function

This routine is used to retrieve and print the abstracts for the hits accumulated for each search. The maximum number of abstracts printed for a search can be restricted so as not to print more abstracts than needed. If the restriction feature is used, then another option exists, the printing of current or historic abstracts.

For each search, control of what could be printed is governed by a control flag. If a specific control flag is not specified then the default control flag is used. The default control flag can be overridden for the entire run if desired. This allows for the maximum amount of flexibility in the printing of the abstracts.

### Program Options

Pagesize	- Change the number of lines per page to a new value from the default of 60.
Col. 1 - 8	- 'Pagesize'
Col. 9 - 10	- NN a number specifying the new page size.
Tapesbeg	- Define the accession number to be regarded as the first on the first tape.
Col. 1 - 8	- 'Tapesbeg'
Col. 10 - 17	- XXXXXXXX The accession number (IE such as ED001001)
Tapesend	- Define the accession number to be regarded as the last on a given

tape. Note that one 'tapesend' card must be present for each tape to be processed.

# VESTAPE

Col. 1 - 8

Control Flag

Col. 1 - 4

Col. 5 - 12

Col. 13 - 16

Col. 17 - 39

- Define that the Vocational Education tape will be processed. If not present the default will be processing of the ERIC tapes.
- 'VESTAPE'
- The printing option can be set for the entire run or for an individual search. The options are the type of search (historic or current) and print requests (suppression or request for a specific entry).
- Search number. If blank the control flag's new setting pertains to the entire run.
- Search type. 'Current' or 'Historic', current is the default.
- Abstract limit. The maximum number of abstracts to be printed per search.
- Print Control Flag. A '1' indicates that the option will be set on, a '0' indicates the option will be set off, and a ' ' indicates no change. These settings are position dependent and defined as follows:

<u>Column</u>	<u>Entry</u>	<u>Position</u>	<u>Default</u>
Col 17	Accession number	0	1
Col 18	Clearing house number	1	0
Col 19	Clearing house number	2	0
Col 20	Program area	3	0
Col 21	Publication date	4	1
Col 22	Title	5	1
Col 23	Personal author	6	1
Col 24	Institution code	7	0
Col 25	Sponsoring agency code	8	0
Col 26	Descriptor	9	1
Col 27	Identifier	10	0
Col 28	EDRS price	11	1
Col 29	Descriptive note	12	1

<u>Column</u>	<u>Entry</u>	<u>Position</u>	<u>Default</u>
Col 30	Issue	13	0
Col 31	Abstract	14	1
Col 32	Report number	15	0
Col 33	Contract number	16	0
Col 34	Grant number	17	0
Col 35	Bureau number	18	0
Col 36	Availability	19	1
Col 37	Journal citation	20	1
Col 38	Institution name	21	0
Col 39	Sponsoring agency name	22	0

NOTE: Position refers to the bit position in the control flag 'control'.

#### Entry Points

- |          |   |
|----------|---|
| ERICABST | - Entry point to program.               |
| EOFIN    | - End of file branch point of RICIN.    |
| EOFHITS  | - End of file branch point for RICHITS. |
| EOFABST  | - End of file branch point for RICABST. |

#### Macros Used

Open  
Close  
Get  
Put  
Time  
Call  
Feov  
Save  
Return

#### Data Sets Used

- |         |   |
|---------|---|
| RICIN   | - This data set contains the control options to the program. Normally this is the card reader.                |
| RICLIST | - This data set contains the printed abstracts that were found for each search. Normally this is the printer. |

## RICHITS

- This data set is used to obtain the next ERIC accession number whose abstract is to be searched for. The format for a record in this data set is defined by the DSECT EDHIT.

## RICABST

- This data set is used to search for the proper abstracts. The form of an entry in a record is in the following form . . 1) 2 BYTE length code - this is the total length of an entry, from the beginning of the length code to the end of the entries information. 2) 2 BYTE entry code - this contains a code that identifies what type of an entry this is. These codes are defined in the VECTOR table 'ENTCODE'. 3) The information for this entry.

## APPENDIX B

### ADDITIONAL HARDWARE CONFIGURATIONS AND COSTS FOR THE MINI-COMPUTER PROGRAM

#### a. Minimal Stand Alone System

This system would require a PDP-8 with 8k memory, a dual DEC tape unit, an IBM compatible magnetic tape drive, a line printer, a teletype and the prerequisites required by this hardware. The following chart shows the costs associated with this hardware configuration over a five year period.

\$44,000	Initial hardware costs.
18,000	Maintenance contract for 57 months
36,000	One shift operator (\$600 per month)
5,000	Overhead
<hr/>	
\$103,000	Total
x 46.5	Per cent of terminal time required to run searches (refer to Chapter 4)
<hr/>	
\$48,000	Total cost for computer time
÷ 6,000	Number of searches over the five year period
<hr/>	
\$ 8.00	Computer time cost per search
+ .30	Cost for paper on which the resumes are printed
<hr/>	
\$ 8.30	Total cost per search on a shared basis

Assuming that a dedicated system was required, a high speed teletype unit could be used in place of the present teletype and line

printer. This would result in the following cost analysis:

\$103,000	Total cost of configuration a.
- 10,000	Reduction in cost when eliminating the line printer
<hr/>	
\$ 93,000	Resultant total
÷ 6,000	Number of searches over the five year period
<hr/>	
\$ 15.50	Total cost per search on a dedicated basis.

b. Ideal Stand-Alone System

The addition of a random access disc to configuration a will result in considerable time savings.

\$103,000	Total cost of configuration a.
7,000	Additional cost for disc with maintenance
<hr/>	
\$110,000	Total
x 38.2	Per cent of terminal time required to run searches (refer to Chapter 3)
<hr/>	
\$ 42,000	Total cost for computer time
÷ 6,000	Number of searches over the five year period
<hr/>	
\$ 7.00	Computer time cost per search
.30	Cost for paper on which the resumes are printed
<hr/>	
\$ 7.30	Total cost per search on a shared basis

Assuming a dedicated system was desired, the faster teletype as described in configuration a would be required.



\$110,000 Total cost of configuration b  
 - 10,000 Reduction in cost when eliminating the line printer  


---

 \$100,000 Resultant total  
 ÷ 6,000 Number of searches over the five year period  


---

 \$ 16.67 Total cost per search on a dedicated basis.

c. Addition of Hardware to Minimal Existing System

Assuming availability of a PDP-8 with 8k memory, dual DEC tape drives, and a teletype on which sufficient time can be obtained in return for additional hardware, the hardware required would be the line printer and magnetic tape drive.

\$ 22,500 Initial hardware costs  
 8,500 Maintenance contract for 57 months  
 24,000 Operator (600 batches x 8.0 hours/batch x \$5/hour)

---

 \$ 55,000 Total cost for computer time  
 ÷ 6,000 Number of searches over the five year period

---

 \$ 9.20 Computer time cost per search  
 + .30 Cost for paper on which the resumes are printed

---

 \$ 9.50 Total cost per search on a dedicated or shared basis

Because the operator costs are dependent upon the amount of operating time, there would be no appreciable cost reduction by printing on a teletype or other slow speed printer.

d. Addition of Hardware to Ideal Existing System

The ideal system would have available a random access disk in addition to the hardware listed under configuration c. The only

difference in the cost analysis of c is the operator time as shown in the following table.

\$ 55,000	Total cost for computer time for configuration c.
- 4,500	Reduction in operator cost (600 batches x 8.0 hours/batch for tape - 6.5 hours/batch for disk - x \$5/hour).
\$ 50,500	Revised total cost for computer time
÷ 6,000	Number of searches over the five year period
\$ 8.40	Computer time cost per search
+ .30	Cost for paper on which the resumes are printed
\$ 8.70	Total cost per search on a dedicated or shared basis.

It should be noted that the cost per search estimates for this and configuration c show the amortization of the hardware charged entirely to the computer search application. Assuming the amortization is spread over all applications that use the hardware and/or that either the line printer or tape drive is already available, a considerable reduction in the cost per search would result.